**I/WE CLAIM:**

1. A graphical user interface framework, the framework comprising:

    a. at least one generic view panel component;

    b. at least one view panel descriptor file, the at least one descriptor file including information specifying the configuration of the at least one generic view panel component;

    c. a parser for run-time loading the at least one descriptor file; and

    d. machine logic coded to selectively configure at run-time the at least one generic view panel component in accordance with the specification detailed in the at least one descriptor file to display at least one view panel instance on a display interface

    wherein the use of the at least one generic view panel component and the run-time loading of the at least one descriptor file by the framework provides for a dynamic configuration of the display interface.

2. A framework as claimed in claim 1, wherein the at least one generic view panel component further comprises at least one widget facilitating human-machine interaction, the run-time configuration of the at least one generic view panel component provides for a dynamic configurable human-machine interface.

3. A framework as claimed in claim 2, wherein information held in the at least one descriptor file further comprises a specification of at least one widget attribute.

4. A framework as claimed in claim 2, wherein information held in the at least one descriptor file further comprises a specification of at least one widget attribute rule.

5. A framework as claimed in claim 4, wherein the at least one widget attribute rule further comprises an indication of at least one allowable action to be performed on the at least one widget.

6. A framework as claimed in claim 2, wherein the at least one generic view panel component includes an error handling routine for processing input.

7. A framework as claimed in claim 1, further comprising a command interpreter for interpreting at least one received command wherein selectively configuring the at least one generic view panel, the machine logic is further responsive to at least one command.

8. A framework as claimed in claim 1, wherein the machine logic further comprises at least one access routine implementation used by the framework to interact with a managed object server brokering access to at least one manageable entity object instance corresponding to a managed data network entity in providing network management and service provisioning solutions.

9. A framework as claimed in claim 8, wherein the framework further comprises messaging means for sending at least one message to the managed object server in interacting therewith.

10. A framework as claimed in claim 9, wherein the at least one message further encapsulates at least one command.

11. A framework as claimed in claim 8, wherein the machine logic further comprises a validate routine implementation used by the framework to validate brokered access to the at least one manageable object entity instance.

12. A framework as claimed in claim 8, wherein the machine logic further comprises a commit routine implementation used by the framework to commit changes to information associated with the at least one managed data network entities instance.

13. A framework as claimed in claim 1, wherein the framework is operable to enable the registration of at least one application module adapted to interwork with the framework, the registration of the at least one application module with the framework provides an independence in the development of the framework as well as in the development of the at least one application module.

14. A framework as claimed in claim 13, wherein the framework further comprises a module registration routine enabling the registration of at least one JAVA plug-in module.

15. A framework as claimed in claim 1, wherein the descriptor files comprise human-readable files.

16. A framework as claimed in claim 15, wherein the descriptor files comprise eXtensible Markup Language (XML) files.

17. A method of providing a human-machine interface, the method comprising steps of:

a. determining an interaction context requiring an instance of a view panel to be displayed, the instance of the view panel comprising at least one view panel component;

b. selecting a descriptor file based on the interaction context to configure at least one view panel component provided by the framework;

c. configuring an instantiated view panel; and

d. displaying the view panel instance

wherein the run-time configuration of view panel components provides for a dynamic configuration of the human-machine interface.

18. A method as claimed in claim 17, wherein the determination of the interaction context requiring the view panel instance, the method comprises a further step of: extracting particulars from a received command using a command interpreter.

19. A method as claimed in claim 18, the method further comprising a step of: making the particulars of the received command available to at least one module.

20. A method as claimed in claim 19, wherein the at least one module comprises a module registered with a graphical user interface framework, the method further comprising the prior step of: registering the module with the framework.

21. A method as claimed in claim 17, wherein the method further comprises a step of accepting input.

22. A method as claimed in claim 21, wherein the method further comprises a step of validating input.

23. A method as claimed in claim 21, wherein the method further comprises a step of committing configuration changes.

24. A method as claimed in claim 17 wherein the method further comprises steps of: exchanging at least one message with a managed object server.

25. A method as claimed in claim 24, wherein exchanging the at least one message with the managed object server, the method further comprises a step of: encapsulating at least one command in the at least one message.

26. A method as claimed in claim 24, wherein exchanging the at least one message with the managed object server, the method further comprises a step of: decapsulating at least one command from a received message.

27. A method as claimed in claim 26, wherein the determination of the interaction context further comprises the step: interpreting a received command.

28. A method as claimed in claim 17, wherein configuring the instantiated view panel, the method further comprises a step of: combining at least two view panel components in instantiating the view panel.